Introduction:

The dataset on aviation accidents from 1908 to 2008 provides a comprehensive historical record of aviation safety incidents over a century. Covering nearly a hundred years, this dataset offers valuable insights into the patterns, trends, and causes of plane crashes, reflecting the evolution of aviation technology and safety measures. By examining data from early aviation pioneers to modern air travel, this dataset enables a detailed analysis of how factors such as aircraft design, operational procedures, and pilot training have impacted safety. It also highlights significant periods of high crash rates, notable advancements in safety, and the evolving nature of aviation risks. This exploration aims to understand historical accident trends, identify key areas for improvement, and inform future strategies to enhance aviation safety and prevent accidents.

In [1]:
```
#Importing Libraries such as pandas,numpy matplotlib and seaborn etc.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
df=pd.read_csv("Airplane_Crashes_and_Fatalities_Since_1908[1].csv") # Impor
ting data inside a varible
```

In [3]:
```
df.shape ##  Shows row and columns
```

Out[3]:
```
(5268, 14)
```

In [4]:
```
df.info() # Give information about the columns and it's data types
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5268 entries, 0 to 5267
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   index         5268 non-null   int64
 1   Date          5268 non-null   object
 2   Time          3049 non-null   object
 3   Location      5248 non-null   object
 4   Operator      5250 non-null   object
 5   Flight #      1069 non-null   object
 6   Route         3561 non-null   object
 7   Type          5241 non-null   object
 8   Registration  4933 non-null   object
 9   cn/In         4040 non-null   object
 10  Aboard        5246 non-null   float64
 11  Fatalities    5256 non-null   float64
 12  Ground        5246 non-null   float64
 13  Summary       4878 non-null   object
dtypes: float64(3), int64(1), object(10)
memory usage: 576.3+ KB
```

In [5]:
```
df.describe() ## Describe function shows statistical data
```

Out[5]:

	index	Aboard	Fatalities	Ground
count	5268.00000	5246.000000	5256.000000	5246.000000
mean	2633.50000	27.554518	20.068303	1.608845
std	1520.88494	43.076711	33.199952	53.987827
min	0.00000	0.000000	0.000000	0.000000
25%	1316.75000	5.000000	3.000000	0.000000
50%	2633.50000	13.000000	9.000000	0.000000
75%	3950.25000	30.000000	23.000000	0.000000
max	5267.00000	644.000000	583.000000	2750.000000

In [6]:
```
df.head() ## It shows top 5 rows of the columns
```

Out[6]:

	index	Date	Time	Location	Operator	Flight #	Route	Type	Registratic
0	0	09/17/1908	17:18	Fort Myer, Virginia	Military - U.S. Army	NaN	Demonstration	Wright Flyer III	Na
1	1	07/12/1912	06:30	AtlantiCity, New Jersey	Military - U.S. Navy	NaN	Test flight	Dirigible	Na
2	2	08/06/1913	NaN	Victoria, British Columbia, Canada	Private	-	NaN	Curtiss seaplane	Na
3	3	09/09/1913	18:30	Over the North Sea	Military - German Navy	NaN	NaN	Zeppelin L-1 (airship)	Na
4	4	10/17/1913	10:30	Near Johannisthal, Germany	Military - German Navy	NaN	NaN	Zeppelin L-2 (airship)	Na

Data Manipulation :

In [7]:
```
#### Checking for Duplicate values
```

In [8]:
```
df.duplicated().sum()    # There is no duplicate values in this dataset
```

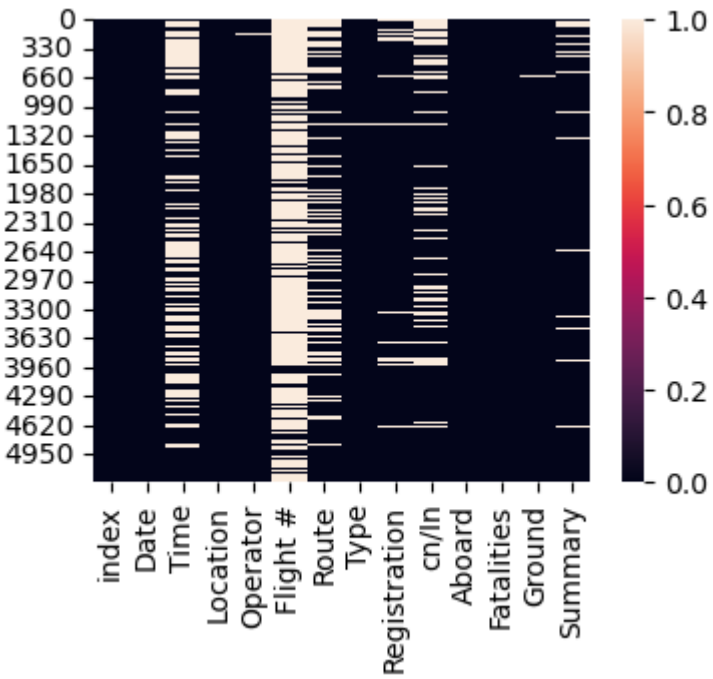
Out[8]: 0

In [9]:
```
df.isnull().sum()  ## Checking null values here
```

Out[9]:
```
index              0
Date               0
Time            2219
Location          20
Operator          18
Flight #        4199
Route           1707
Type              27
Registration     335
cn/In           1228
Aboard            22
Fatalities        12
Ground            22
Summary          390
dtype: int64
```

In [10]:
```
## Using Heatmap to check null values by Visualiziton
```

In [11]:
```
plt.figure(figsize=(4,3))
sns.heatmap(df.isnull())    # White lines shows there are null values perse
nt in these columns
plt.show()
```



In [12]:
```
(df.isnull().sum()/df.shape[0])*100  ## checking null values in each column
s in percentage
```

Out[12]:
```
index            0.000000
Date             0.000000
Time            42.122248
Location         0.379651
Operator         0.341686
Flight #        79.707669
Route           32.403189
Type             0.512528
Registration     6.359150
cn/In           23.310554
Aboard           0.417616
Fatalities       0.227790
Ground           0.417616
Summary          7.403189
dtype: float64
```

In [13]:
```
## Dropping Unwanted columns like index,Registration,cn/In and those column
s that contains null values more than 70% as like Flight #.
```

In [14]:
```
df.drop(["index","Flight #","Registration","cn/In"],axis=1,inplace=True)
```

In [15]:
```
df.columns  ## These are the remaining columns that we are going to analyz
e.
```

Out[15]:
```
Index(['Date', 'Time', 'Location', 'Operator', 'Route', 'Type', 'Aboard',
       'Fatalities', 'Ground', 'Summary'],
      dtype='object')
```

In [16]:
```
## 1- We are creating new columns here .
```

In [17]:
```
df["Year"]=pd.to_datetime(df["Date"]).dt.year
```

In [18]:
```
df["Year"]
```

Out[18]:
```
0       1908
1       1912
2       1913
3       1913
4       1913
        ...
5263    2009
5264    2009
5265    2009
5266    2009
5267    2009
Name: Year, Length: 5268, dtype: int32
```

In [19]:
```
df["Crash_Region"]=df["Location"].str.split(",").str[-1]
```

In [20]:
```
df["Crash_Region"]
```

Out[20]:
```
0                     Virginia
1                   New Jersey
2                       Canada
3           Over the North Sea
4                      Germany
                 ...
5263                 Indonesia
5264     DemocratiRepubliCongo
5265                    Brazil
5266                    Canada
5267                     India
Name: Crash_Region, Length: 5268, dtype: object
```

In [21]:
```
df["Total_Fatalities"]=df["Fatalities"]+df["Ground"]
```

In [22]:
```
df["Total_Fatalities"]
```

Out[22]:
```
0         1.0
1         5.0
2         1.0
3        14.0
4        30.0
        ...
5263    100.0
5264      NaN
5265    228.0
5266      1.0
5267     13.0
Name: Total_Fatalities, Length: 5268, dtype: float64
```

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5268 entries, 0 to 5267
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Date              5268 non-null   object
 1   Time              3049 non-null   object
 2   Location          5248 non-null   object
 3   Operator          5250 non-null   object
 4   Route             3561 non-null   object
 5   Type              5241 non-null   object
 6   Aboard            5246 non-null   float64
 7   Fatalities        5256 non-null   float64
 8   Ground            5246 non-null   float64
 9   Summary           4878 non-null   object
 10  Year              5268 non-null   int32
 11  Crash_Region      5248 non-null   object
 12  Total_Fatalities  5246 non-null   float64
dtypes: float64(4), int32(1), object(8)
memory usage: 514.6+ KB
```

In [24]:
```
## We are going to change data types of "Date" and "Time" columns.
```

In [25]:
```
df["Date"]=pd.to_datetime(df["Date"],errors="coerce")
```

In [26]:
```
df["Time"]=pd.to_datetime(df["Time"],errors="coerce")
```

In [27]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5268 entries, 0 to 5267
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Date              5268 non-null   datetime64[ns]
 1   Time              3036 non-null   datetime64[ns]
 2   Location          5248 non-null   object
 3   Operator          5250 non-null   object
 4   Route             3561 non-null   object
 5   Type              5241 non-null   object
 6   Aboard            5246 non-null   float64
 7   Fatalities        5256 non-null   float64
 8   Ground            5246 non-null   float64
 9   Summary           4878 non-null   object
 10  Year              5268 non-null   int32
 11  Crash_Region      5248 non-null   object
 12  Total_Fatalities  5246 non-null   float64
dtypes: datetime64[ns](2), float64(4), int32(1), object(6)
memory usage: 514.6+ KB
```

In [28]:
```
## Going to handle missing values.
```

In [29]:
```
(df.isnull().sum()/df.shape[0])*100
```

Out[29]:
```
Date                 0.000000
Time                42.369021
Location             0.379651
Operator             0.341686
Route               32.403189
Type                 0.512528
Aboard               0.417616
Fatalities           0.227790
Ground               0.417616
Summary              7.403189
Year                 0.000000
Crash_Region         0.379651
Total_Fatalities     0.417616
dtype: float64
```

In [30]:
```
df["Time"].mean().strftime("%Y:%m:%d %H:%M")
```

Out[30]:
```
'2024:08:25 13:16'
```

In [31]:
```
df["Time"]=df["Time"].fillna(df["Time"].mean().strftime("%Y:%m:%d %H:%M"))
```

In [32]:
```
df["Route"]=df["Route"].fillna(method="bfill")
```

In [33]:
```
df["Summary"].mode()[0]
```

Out[33]:
```
'Crashed during takeoff.'
```

In [34]:
```
df["Summary"]=df["Summary"].fillna(df["Summary"].mode()[0])
```

In [35]:
```
df.dropna(inplace=True)
```

In [36]:
```
df.isnull().sum()
```

Out[36]:
```
Date                0
Time                0
Location            0
Operator            0
Route               0
Type                0
Aboard              0
Fatalities          0
Ground              0
Summary             0
Year                0
Crash_Region        0
Total_Fatalities    0
dtype: int64
```

In [37]:
```
df.head(1)
```

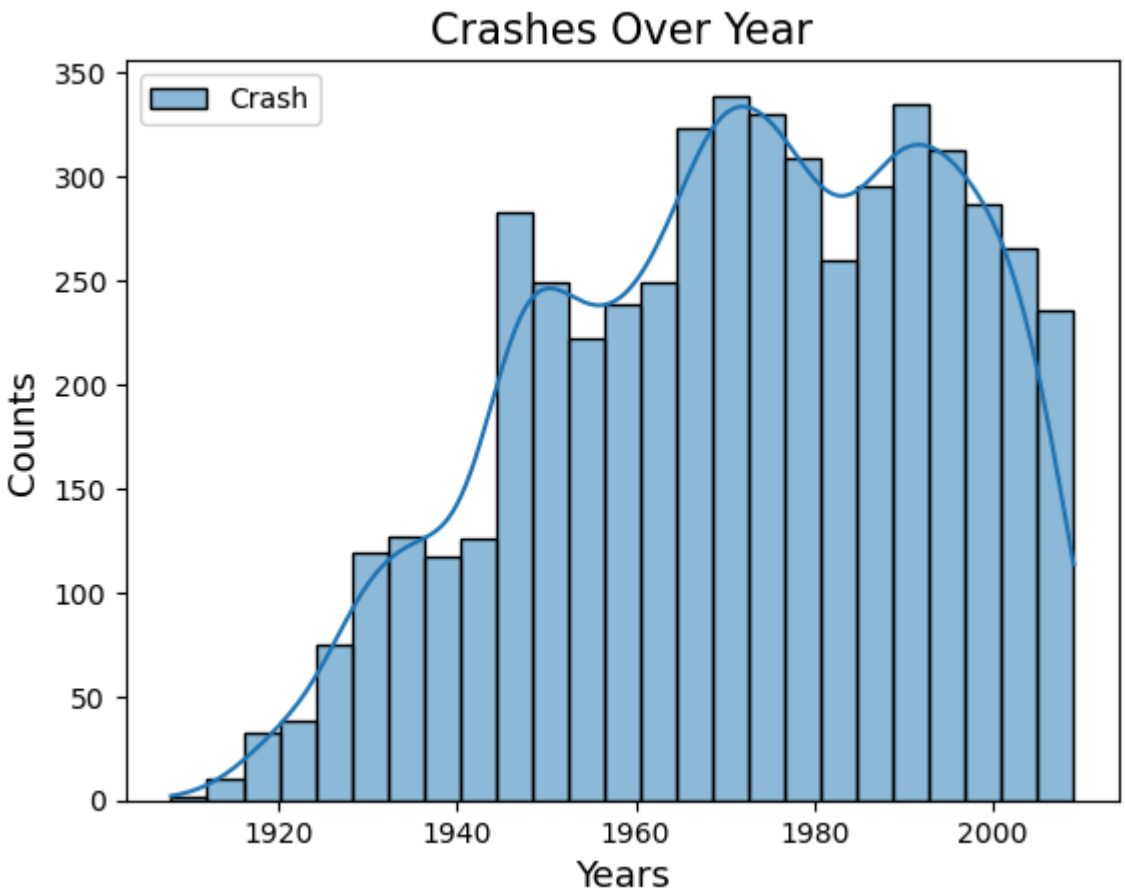
Out[37]:

	Date	Time	Location	Operator	Route	Type	Aboard	Fatalities	Ground	
0	1908-09-17	2024-08-25 17:18:00	Fort Myer, Virginia	Military - U.S. Army	Demonstration	Wright Flyer III	2.0	1.0	0.0	dem fli

Exploratory Data Analysis:

Crashes over Year:

In [38]:
```
## Plotting a Histogram
sns.histplot(x=df["Year"],kde=True,label="Crash")
plt.legend()
plt.xlabel("Years",size=13)
plt.ylabel("Counts",size=13)
plt.title("Crashes Over Year",size=15)
plt.show()
```



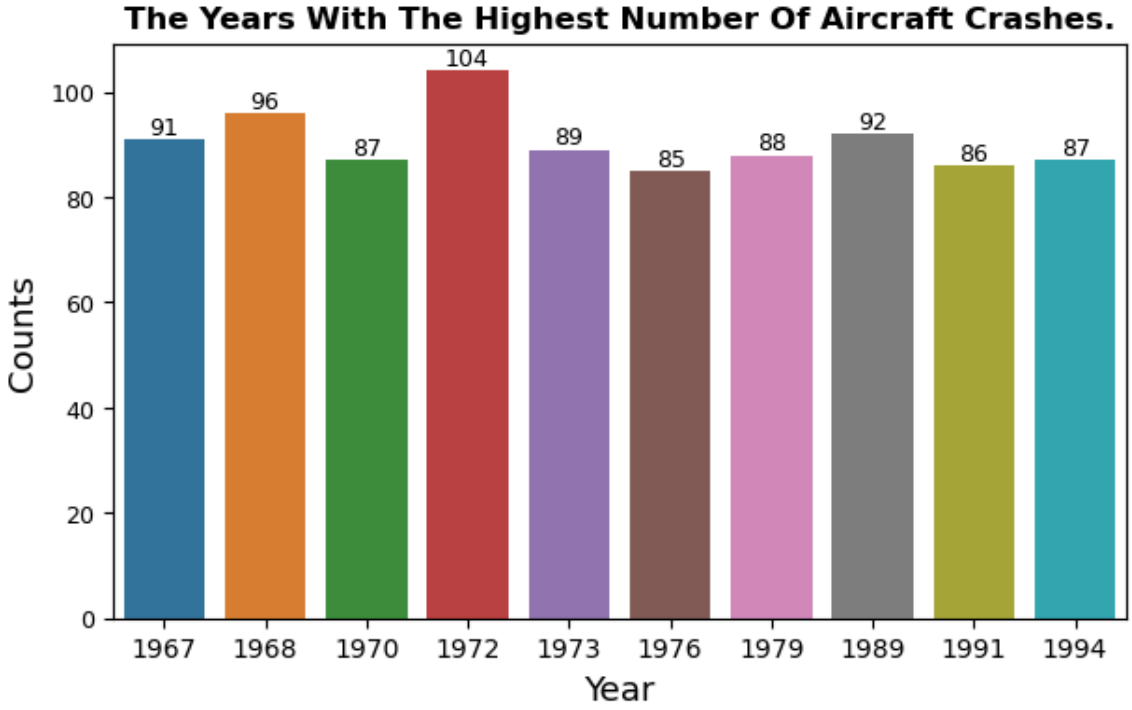
As we can see from the above graph,most of the crashes occured between 1942 and 2000.

The top 10 years with the highest number of Aircraft Crashes:

In [39]:
```
year_index=df["Year"].value_counts().head(10).index
```

In [40]:
```
year_values=df["Year"].value_counts().head(10).values
```

In [41]:
```
plt.figure(figsize=(7,4))
az=sns.barplot(x=year_index,y=year_values,saturation=.7)
for bars in az.containers:
    az.bar_label(bars,size=9)
plt.ylabel("Counts",size=13)
plt.xlabel("Year",size=13)
plt.xticks(size=10)
plt.yticks(size=9)
plt.title("The Years With The Highest Number Of Aircraft Crashes.",fontweig
ht='bold')
plt.show()
```



"The maximum number of plane crashes occurred in 1972 with 104 crashes, followed by 1968 with 96 crashes."

"Top 10 years with the highest number of fatalities:"

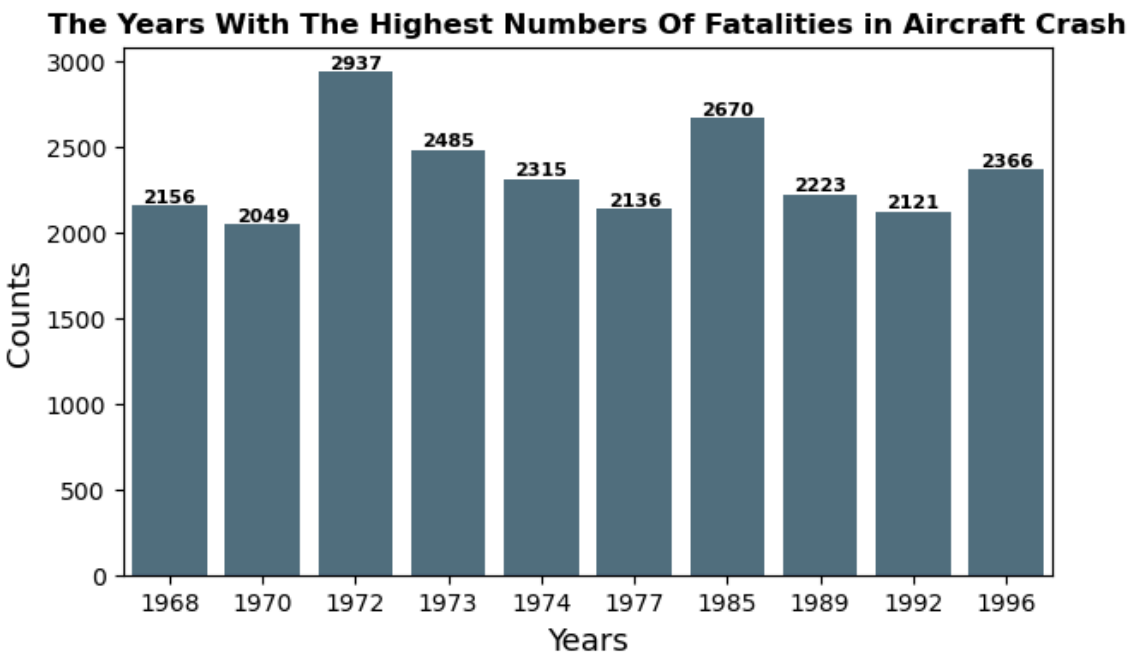
In [42]:
```
death_rate=df.groupby(["Year"],as_index=False)["Fatalities"].sum().sort_val
ues(by=["Fatalities"],ascending=False).head(10)
```

In [43]:
```
death_rate
```

Out[43]:

	Year	Fatalities
60	1972	2937.0
73	1985	2670.0
61	1973	2485.0
84	1996	2366.0
62	1974	2315.0
77	1989	2223.0
56	1968	2156.0
65	1977	2136.0
80	1992	2121.0
58	1970	2049.0

In [44]:
```
plt.figure(figsize=(7,4))
vv=sns.barplot(x="Year",y="Fatalities",data=death_rate,color="#1f78b4",satu
ration=.3)
for bars in vv.containers:
    vv.bar_label(bars,size=8,fontweight='bold')
plt.xlabel("Years",size=13)
plt.ylabel("Counts",size=13)
plt.title("The Years With The Highest Numbers Of Fatalities in Aircraft Cra
sh",fontweight='bold')
plt.show()
```



"The maximum number of fatalities occurred in 1972 with 2,937 fatalities, followed by 1985 with 2,670 fatalities."

Year-wise Aboard and Fatalities using Line Graph:

In [45]:
```
time_series=df.groupby(["Year"],as_index=False)[["Fatalities","Aboard"]].su
m()
```

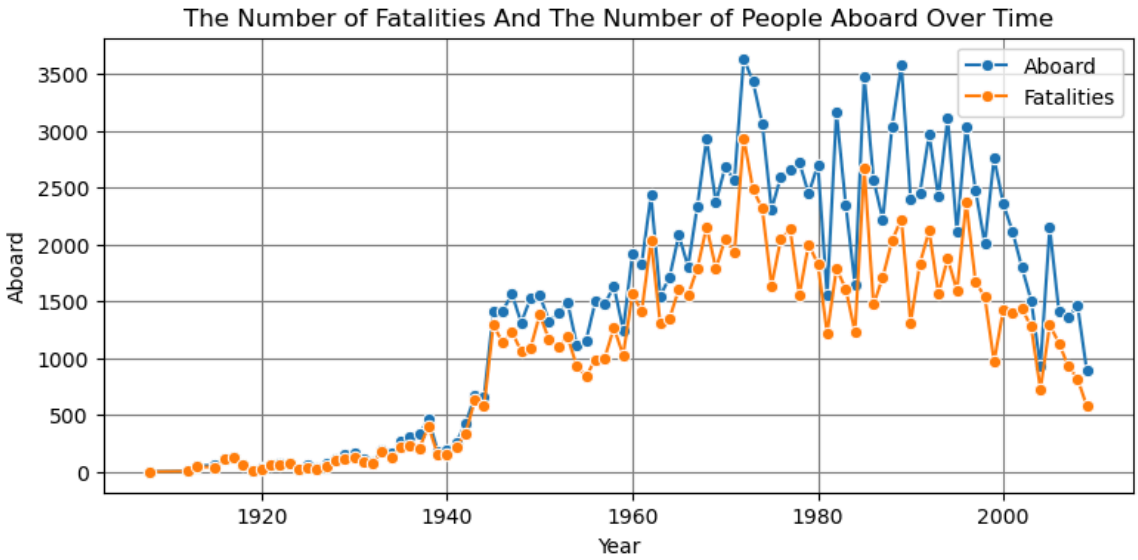
In [46]:
```
time_series
```

Out[46]:

	Year	Fatalities	Aboard
0	1908	1.0	2.0
1	1912	5.0	5.0
2	1913	45.0	51.0
3	1915	40.0	60.0
4	1916	108.0	109.0
...	...	...	...
93	2005	1291.0	2146.0
94	2006	1132.0	1408.0
95	2007	927.0	1360.0
96	2008	820.0	1463.0
97	2009	577.0	887.0

98 rows × 3 columns

In [47]:
```
# Plotting a Line Graph:
plt.figure(figsize=(9,4))
sns.lineplot(x="Year",y="Aboard",data=time_series,label="Aboard",marker
="o")
sns.lineplot(x="Year",y="Fatalities",data=time_series,label="Fatalities",ma
rker="o")
plt.title("The Number of Fatalities And The Number of People Aboard Over Ti
me")
plt.grid(color="grey")
plt.show()
```

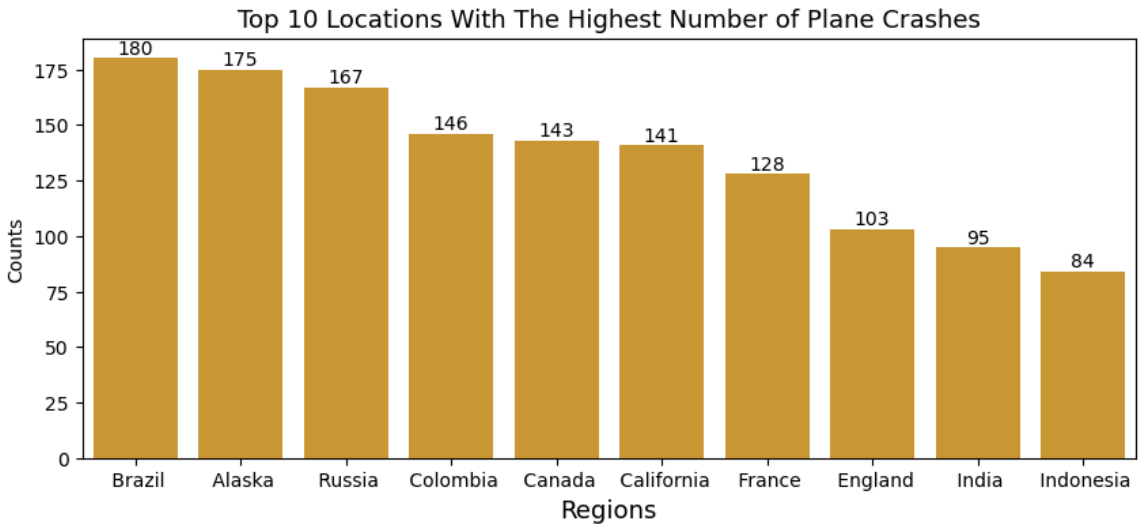


Top 10 Region with Higheste Number of Crashes:

In [48]:
```
index_region=df["Crash_Region"].value_counts().head(10).index
```

In [49]:
```
value_region=df["Crash_Region"].value_counts().head(10).values
```

In [50]:
```
plt.figure(figsize=(10,4))
ss=sns.barplot(x=index_region,y=value_region,color="orange",saturation=.6)
for bars in ss.containers:
    ss.bar_label(bars)
plt.xlabel("Regions",size=13)
plt.ylabel("Counts")
plt.title("Top 10 Locations With The Highest Number of Plane Crashes",size=
13)
plt.show()
```



"Brazil has the highest number of plane crashes with 180 incidents, followed by Alaska with 175 and Russia with 167."

In [51]:
```
## Reason of the Plane Crashes in Brazil
df[(df["Location"]=="Sao Paulo, Brazil")][["Summary"]].head()
```

Out[51]:

	Summary
469	The mail plane crashed while taking off.
664	Crashed in fog.
836	Crashed into the Solimoes extension of the Ama...
1148	Crashed into a house shortly after taking off ...
1203	Crashed while attempting to make an emergency ...

In [52]:
```
## Reason of the Plane Crashes in Alaska
df[(df["Location"]=="Anchorage, Alaska")][["Summary"]].head()
```

Out[52]:

	Summary
2064	Fatigue fracture on right wing leading to infl...
2437	The military charter overran the runway during...
2481	The aircraft took off from a roadside lodging,...
2864	Crashed short of the runway in fog. The pilot ...
2950	The cargo plane crashed while attempting to ta...

In [53]:
```
## Reason of the Plane Crashes in Russia
df[(df["Location"]=="Moscow, Russia")][["Summary"]].head()
```

Out[53]:

	Summary
1684	Crashed on approach to Moscow, 11 nm short of ...
2089	Crashed into a snowbank on the takeoff roll in...
2216	Struck power lines while landing.
2335	Engine fire led to an emergency landing with t...
2500	Crashed on takeoff

"As we can see, most plane crashes occurred during takeoff. The reasons for these crashes may include bad weather,fogs,enigine failure, technical errors,collide with mountains,old planes,poor runway,Hijack,war, or pilot's mistakes etc."

Top 10 Regions with Highest Number of Fatalities:

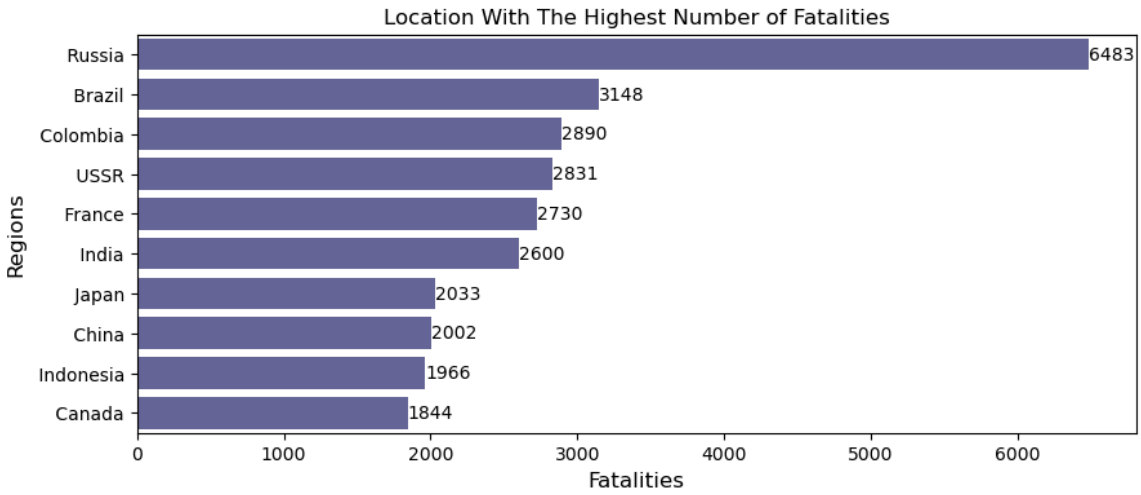
In [54]:
```
death_rate=df.groupby(["Crash_Region"],as_index=False)["Fatalities"].sum().
sort_values(by=["Fatalities"],ascending=False).head(10)
```

In [55]:
```
death_rate
```

Out[55]:

	Crash_Region	Fatalities
315	Russia	6483.0
73	Brazil	3148.0
101	Colombia	2890.0
374	USSR	2831.0
144	France	2730.0
182	India	2600.0
197	Japan	2033.0
99	China	2002.0
186	Indonesia	1966.0
88	Canada	1844.0

In [56]:
```
plt.figure(figsize=(10,4))
qq=sns.barplot(y="Crash_Region",x="Fatalities",data=death_rate,color='blu
e',saturation=.2)
for bars in qq.containers:
    qq.bar_label(bars)
plt.xlabel("Fatalities",size=12)
plt.ylabel("Regions",size=12)
plt.title("Location With The Highest Number of Fatalities")
plt.show()
```



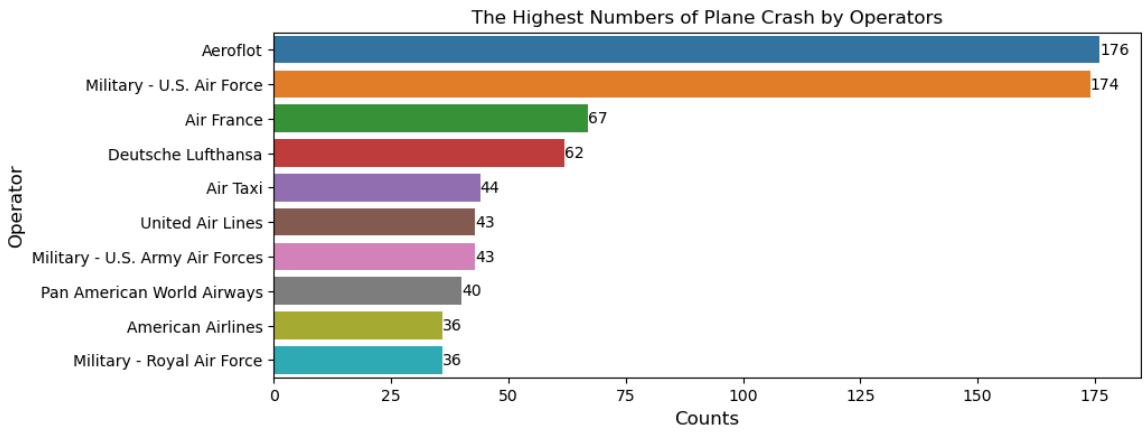
" Russia had the highest number of fatalities with 6483, followed by Brazil with 3148 and Colombia with 2890."

Top 10 Operator with Highest Number of Crashes:

In [57]:
```
indexop_crash=df["Operator"].value_counts().head(10).index
```

In [58]:
```
valueop_crash=df["Operator"].value_counts().head(10).values
```

In [59]:
```
plt.figure(figsize=(10,4))
xx=sns.barplot(y=indexop_crash,x=valueop_crash)
for bars in xx.containers:
    xx.bar_label(bars)
plt.ylabel("Operator",size=12)
plt.xlabel("Counts",size=12)
plt.title("The Highest Numbers of Plane Crash by Operators",size=12)
plt.show()
```



"We can see in the graph above that 'Aeroflot' had 176 crashes, followed by the Military-U.S. Air Force with 174."

In [60]:
```
df[(df["Operator"]=="Aeroflot")][["Summary"]].value_counts().head()
```

Out[60]:
```
Summary
Crashed during takeoff.                                          15
Crashed during approach.                                          4
Crashed on approach.                                              3
Crashed shortly after taking off.                                 2
The aircraft struck a mountain after an attempted go-around.      1
Name: count, dtype: int64
```

In [61]:
```
df[(df["Operator"]=="Military - U.S. Air Force")][["Summary"]].value_counts
().head()
```

Out[61]:
```
Summary
Crashed during takeoff.
6
Struck a mountain.
3
Shot down by enemy fire.
2
Crashed and burned while attempting to land.
2
A course deviation led to the aircraft crashing into Mt. McKinley at an el
evation of 12,000 ft.    1
Name: count, dtype: int64
```

"As we can see, most plane crashes occurred during takeoff. The reasons for these crashes may include bad weather,fogs,enigine failure, technical errors,collide with mountains,old planes,poor runway,Hijack,war, or pilot's mistakes etc."

Top 10 Operator with the Highest Number of Fatalities:

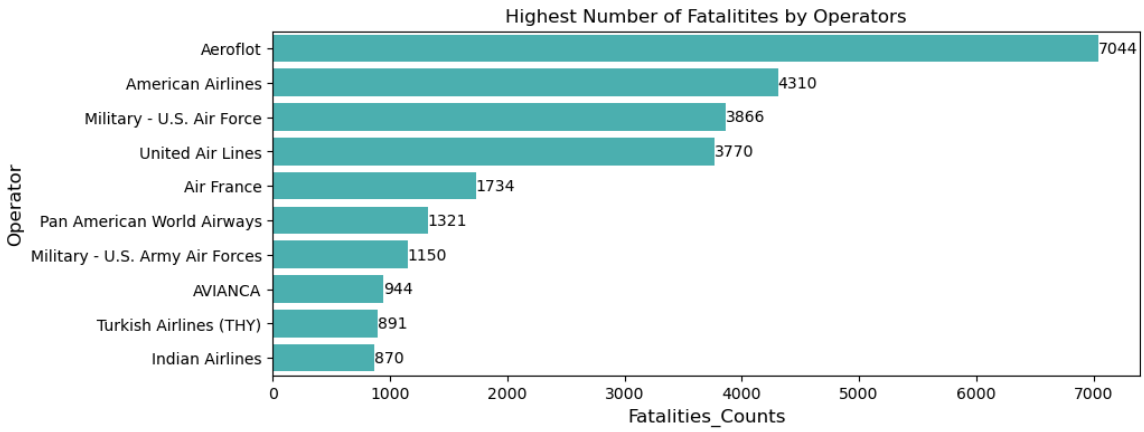
In [62]:
```
totalop_fatalities=df.groupby(["Operator"],as_index=False)["Total_Fatalitie
s"].sum().sort_values(by=["Total_Fatalities"],ascending=False).head(10)
```

In [63]:
```
totalop_fatalities
```

Out[63]:

	Operator	Total_Fatalities
83	Aeroflot	7044.0
431	American Airlines	4310.0
1554	Military - U.S. Air Force	3866.0
2335	United Air Lines	3770.0
196	Air France	1734.0
1750	Pan American World Airways	1321.0
1566	Military - U.S. Army Air Forces	1150.0
22	AVIANCA	944.0
2306	Turkish Airlines (THY)	891.0
1122	Indian Airlines	870.0

In [64]:
```
# Plotting a Bar Graph:
plt.figure(figsize=(10,4))
ww=sns.barplot(y="Operator",x="Total_Fatalities",data=totalop_fatalities,sa
turation=.4,color="cyan")
for bars in ww.containers:
    ww.bar_label(bars)
plt.xlabel("Fatalities_Counts",size=12)
plt.ylabel("Operator",size=12)
plt.title("Highest Number of Fatalitites by Operators",size=12)
plt.show()
```



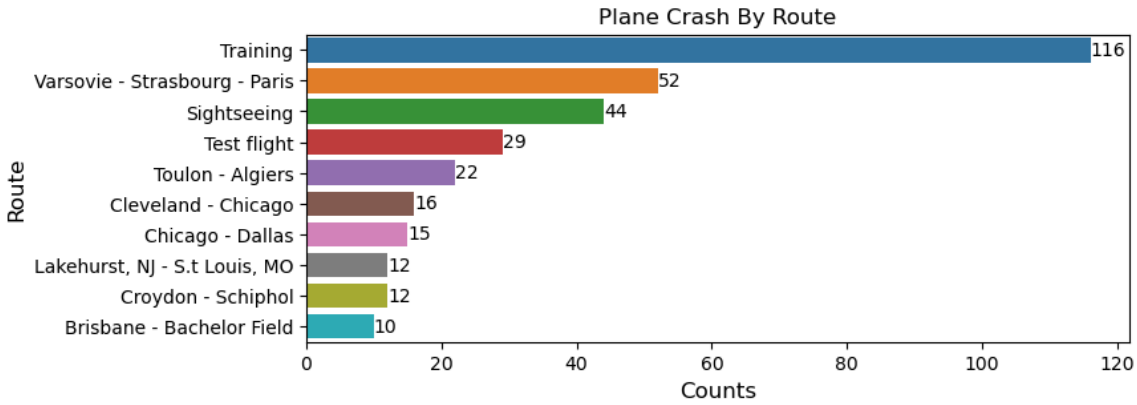
"We can see in the graph above that 'Aeroflot' had 7044 crashes, followed by American Airlines with 4310 and the Military-U.S. Air Force with 3770."

Top 10 Route with Highest number of Crashes:

In [65]:
```
index_route=df["Route"].value_counts().head(10).index
```

In [66]:
```
value_route=df["Route"].value_counts().head(10).values
```

In [67]:
```
plt.figure(figsize=(8,3))
dd=sns.barplot(x=value_route,y=index_route)
for bars in dd.containers:
    dd.bar_label(bars)
plt.xlabel("Counts",size=12)
plt.ylabel("Route",size=12)
plt.title("Plane Crash By Route")
plt.show()
```



"In the above figure, the highest number of plane crashes occurred on the Training route, with 116 crashes."

Top 10 Routes with the Highest Fatalities:

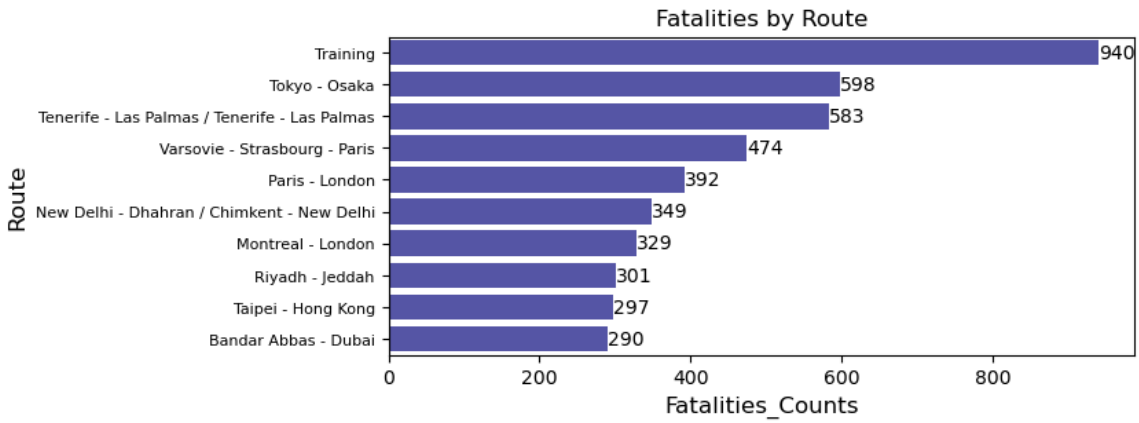
In [68]:
```
route_fatalities=df.groupby(["Route"],as_index=False)["Fatalities"].sum().s
ort_values(by=["Fatalities"],ascending=False).head(10)
```

In [69]:
```
route_fatalities
```

Out[69]:

	Route	Fatalities
3015	Training	940.0
2998	Tokyo - Osaka	598.0
2954	Tenerife - Las Palmas / Tenerife - Las Palmas	583.0
3101	Varsovie - Strasbourg - Paris	474.0
2267	Paris - London	392.0
2049	New Delhi - Dhahran / Chimkent - New Delhi	349.0
1950	Montreal - London	329.0
2513	Riyadh - Jeddah	301.0
2883	Taipei - Hong Kong	297.0
230	Bandar Abbas - Dubai	290.0

In [70]:
```
plt.figure(figsize=(7,3))
cc=sns.barplot(y="Route",x="Fatalities",data=route_fatalities,color="blue",
saturation=.3)
for bar in cc.containers:
    cc.bar_label(bar,color="black")
plt.yticks(size=8)
plt.ylabel("Route",size=12)
plt.xlabel("Fatalities_Counts",size=12)
plt.title("Fatalities by Route")
plt.show()
```



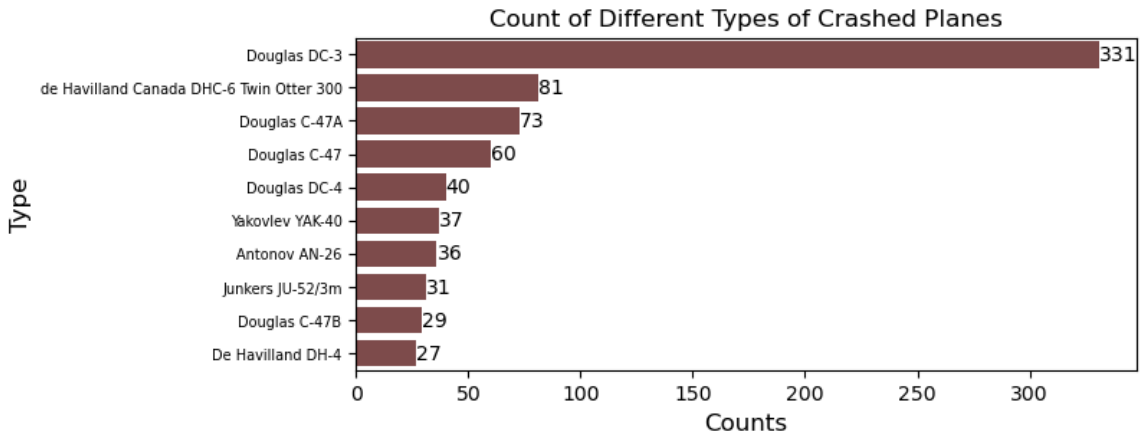
"In the above figure, the highest number of fatalities occurred on the Training route, with 940 fatalities, followed by the Tokyo-Osaka route with 598 fatalities."

Types of Crash Plane:

In [71]:
```
index_type=df["Type"].value_counts().head(10).index
```

In [72]:
```
value_type=df["Type"].value_counts().head(10).values
```

In [73]:
```
plt.figure(figsize=(7,3))
bb=sns.barplot(y=index_type,x=value_type,color="brown",saturation=.4)
for bars in bb.containers:
    bb.bar_label(bars)
plt.yticks(size=7)
plt.ylabel("Type",size=12)
plt.xlabel("Counts",size=12)
plt.title("Count of Different Types of Crashed Planes",size=12)
plt.show()
```



"From the above figure, we can see that the Douglas DC-3 type of plane had 331 crashes."

In [74]:
```
df[(df["Type"]=="Douglas DC-3")][["Summary"]].value_counts().head()
```

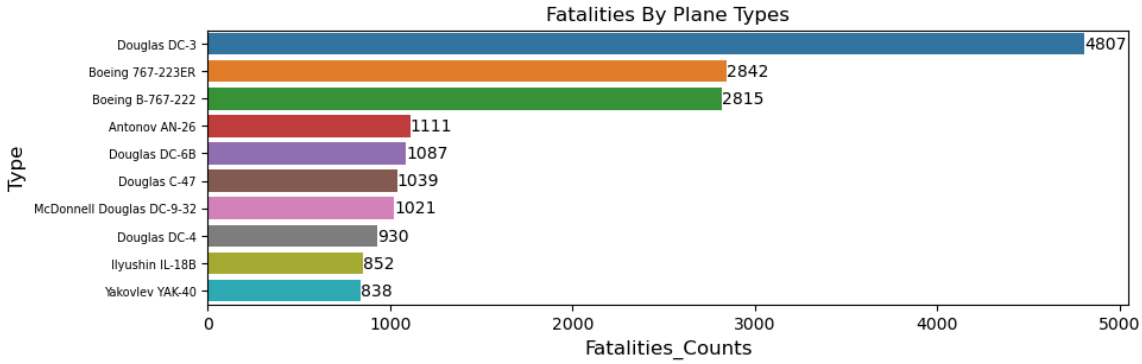
Out[74]:
```
Summary
Crashed during takeoff.    19
Crashed while en route.     3
Flew into a mountain.       3
Crashed en route.           3
Crashed on takeoff.         2
Name: count, dtype: int64
```

"As we can see, most plane crashes occurred during takeoff. The reasons for these crashes may include bad weather, technical errors, old planes,poor runway or pilot's mistakes."

Types of Plane, and Fatalities:

In [75]:
```
total_type=df.groupby(["Type"],as_index=False)["Total_Fatalities"].sum().so
rt_values(by=["Total_Fatalities"],ascending=False).head(10)
```

In [76]:
```
plt.figure(figsize=(10,3))
tt=sns.barplot(y="Type",x="Total_Fatalities",data=total_type)
for bars in tt.containers:
    tt.bar_label(bars)
plt.xlabel("Fatalities_Counts",size=12)
plt.ylabel("Type",size=12)
plt.title("Fatalities By Plane Types")
plt.yticks(size=7)
plt.show()
```



"As we can see in the graph above, the Douglas DC-3 had the most fatalities with 4807, followed by the Boeing 767-223ER."

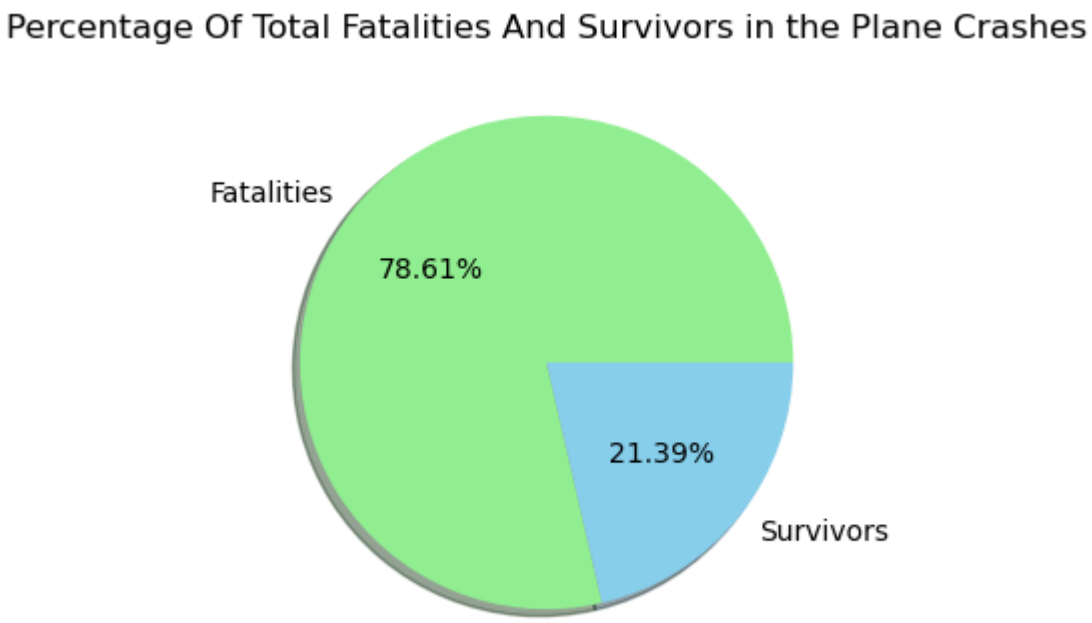
Total Survivors:

In [118]:
```
total_aboard=df["Aboard"].sum()
total_fatalities=df["Total_Fatalities"].sum()
total_survivors=total_aboard-total_fatalities
total_survivors
```

Out[118]: 30752.0

In [119]:
```
sizes=[total_fatalities,total_survivors]
labels=["Fatalities","Survivors"]
colors=["lightgreen","skyblue"]
```

In [120]:
```
# Plotting a Pie Chart for total Survivors
plt.figure(figsize=(8, 4))
plt.pie(sizes, labels=labels, colors=colors, autopct='%.2f%%', shadow=True)
plt.title("Percentage Of Total Fatalities And Survivors in the Plane Crashe
s")
plt.show()
```



"As we can see in the above figure, only 21.39%(30752 in numbers) of people survived in plane crash accidents, while 78.61%(113039 in numbers) died."

Plane Crashed By Hijackers:

In [80]:
```
index_year=df[(df["Summary"].str.contains("Hijacked",case=False,na=False))]
["Year"].value_counts().index
```

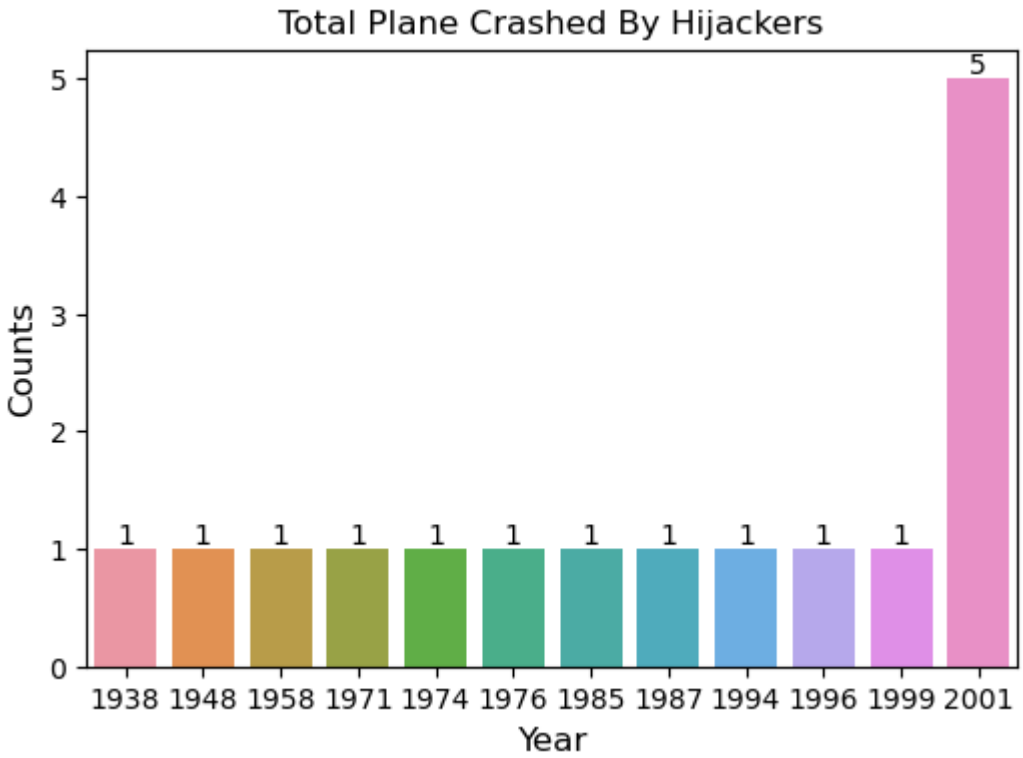
In [81]:
```
value_year=df[(df["Summary"].str.contains("Hijacked",case=False,na=False))]
["Year"].value_counts().values
```

In [82]:
```
df[(df["Summary"].str.contains("Hijack",case=False,na=False))][["Summar
y","Year"]].head()
```

Out[82]:

	Summary	Year
480	The plane crashed into the ocean while en rout...	1938
953	The flight crashed after being hijacked and lo...	1948
1235	After takeoff from Laoag an armed man forced h...	1952
1568	Hijacked by 4 Cuban rebels, the plane crashed ...	1958
2455	Crash landed on a beach after a hijacker deton...	1971

In [83]:
```
plt.figure(figsize=(6,4))
bn=sns.barplot(x=index_year,y=value_year)
for bar in bn.containers:
    bn.bar_label(bar)
plt.xlabel("Year",size=12)
plt.ylabel("Counts",size=12)
plt.title("Total Plane Crashed By Hijackers")
plt.show()
```



"According to the figure above, the highest number of plane crashes occurred in 2001, with 5 incidents."

Military Plane Crashes Over Time:

In [84]:
```
## Military Plane Crashes
```

In [85]:
```
military_operator=df[(df["Operator"].str.contains("Military",case=False,na=
False))]
```

In [86]:
```
military_plot=military_operator.groupby(["Year"],as_index=False)["Operato
r"].count().sort_values(by=["Operator"],ascending=False)
```

In [87]:
```
military_plot
```

Out[87]:

	Year	Operator
21	1945	44
22	1946	25
44	1968	22
43	1967	18
48	1972	16
...	...	...
13	1933	1
15	1939	1
16	1940	1
35	1959	1
0	1908	1

86 rows × 2 columns

In [88]:
```
### Civil Plane Crashes
```

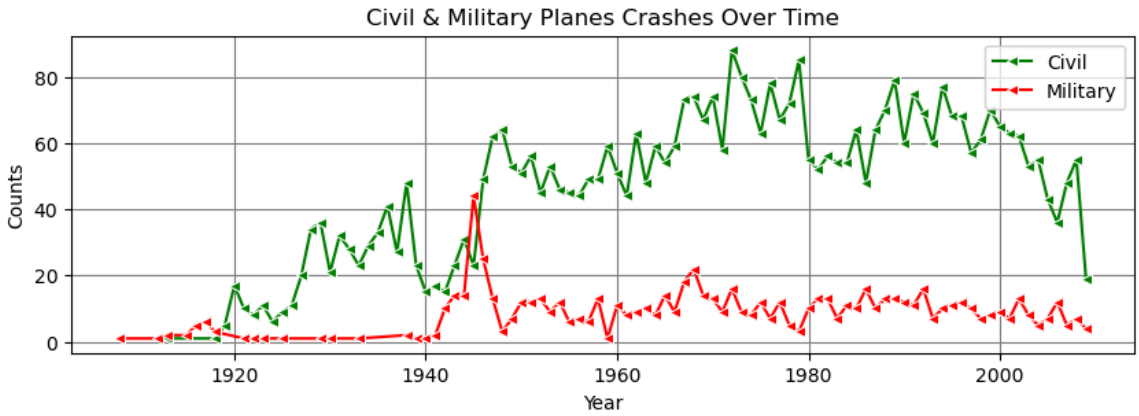
In [89]:
```
mask = df["Operator"].str.contains("Military", case=False, na=False)
df_filtered=df[~mask]
civil_plane=df_filtered.groupby(["Year"],as_index=False)["Operator"].count
().sort_values(by=["Operator"],ascending=False)
civil_plane
```

Out[89]:

	Year	Operator
55	1972	88
62	1979	85
56	1973	80
72	1989	79
59	1976	78
...	...	...
5	1922	8
7	1924	6
2	1919	5
1	1918	1
0	1913	1

93 rows × 2 columns

In [90]:
```
# Plotting a Line Graph
plt.figure(figsize=(10,3))
sns.lineplot(x="Year",y="Operator",data=civil_plane,marker="<",color="g",la
bel="Civil")
sns.lineplot(x="Year",y="Operator",data=military_plot,marker="<",color="r",
label="Military")
plt.title("Civil & Military Planes Crashes Over Time")
plt.ylabel("Counts")
plt.xlabel("Year")
plt.grid(color='grey')
plt.show()
```



1- We can see in the figure above that the highest number of military plane crashes occurred in 1942, with around 44 incidents.

2- The maximum number of Civil Plane Crashes in 1972 was around 85 incidents.

3- Compared to military plane crashes, civil plane crashes occurred more frequently.

In [91]:
```
df[(df["Year"]==1945)][["Summary"]].head(10)  ### Reason of the Crashes
```

Out[91]:

	Summary
670	Crashed during takeoff.
687	The cargo plane crashed in strong winds
688	The aircraft crashed 1.25 miles short of the i...
689	The aircraft, lost in fog, crashed into the Ve...
690	The cargo plane struck a mountain.
691	Crashed during takeoff.
692	Crashed during takeoff.
693	Crashed during takeoff.
694	Crashed into a hill after encountering a fog b...
695	Struck a mountain while flying in low clouds a...

"As we can see, most plane crashes occurred during takeoff. The reasons for these crashes may include bad weather, technical errors, old planes,poor runway or pilot's mistakes."

Crash Percentage of Civil vs. Military Planes:

In [92]:
```
military_operator=df[(df["Operator"].str.contains("Military",case=False,na=
False))].value_counts().values.sum()
military_operator
```

Out[92]: 768

In [93]:
```
total_operator=df["Year"].value_counts().values.sum()
total_operator
```

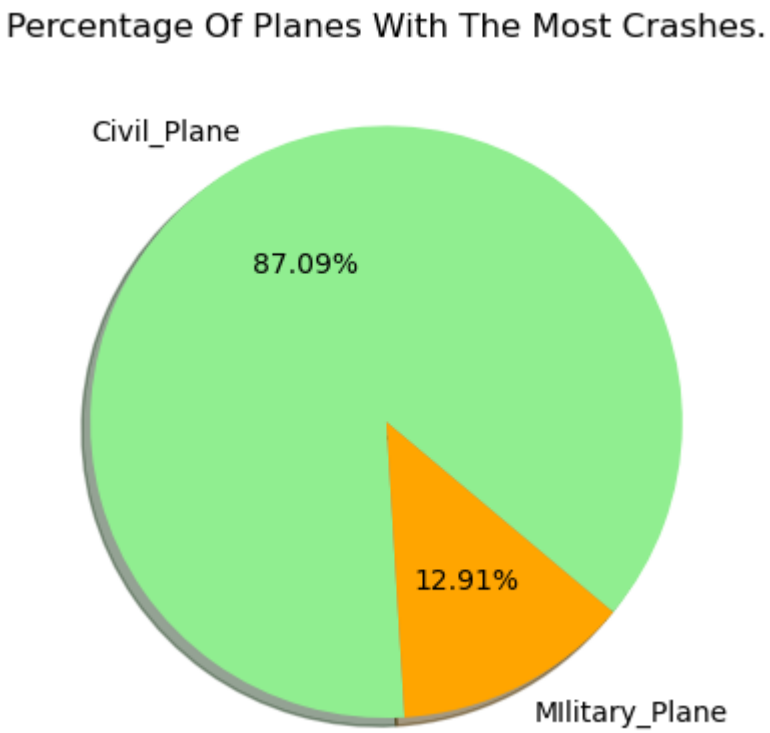
Out[93]: 5181

In [94]:
```
civil=total_operator-military_operator
civil
```

Out[94]: 4413

In [95]:
```
sizes=[total_operator,military_operator]
labels=["Civil_Plane","MIlitary_Plane"]
colors=["lightgreen","orange"]
```

In [96]:
```
# Plotting a Pie Chart:
plt.pie(sizes,labels=labels,colors=colors,autopct="%.2f%%",shadow=True,star
tangle=320)
plt.title("Percentage Of Planes With The Most Crashes.")
plt.show()
```



In the above figure,87.09% of the crashes were civil planes, while 12.91% were military planes.

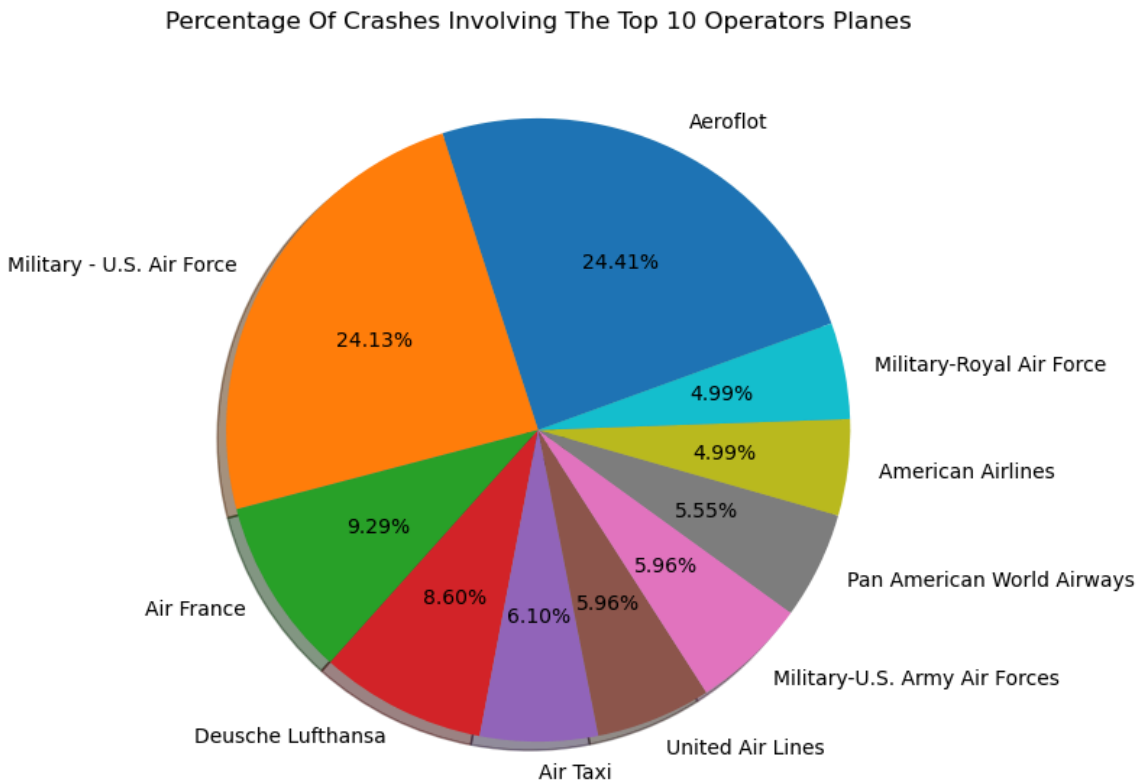
Percentage of the top 10 planes involved in crashes:

In [113]:
```
dist=df["Operator"].value_counts().head(10)
dist
```

Out[113]:
```
Operator
Aeroflot                            176
Military - U.S. Air Force           174
Air France                           67
Deutsche Lufthansa                   62
Air Taxi                             44
United Air Lines                     43
Military - U.S. Army Air Forces      43
Pan American World Airways           40
American Airlines                    36
Military - Royal Air Force           36
Name: count, dtype: int64
```

In [114]:
```
labels=["Aeroflot","Military - U.S. Air Force","Air France","Deusche Luftha
nsa","Air Taxi","United Air Lines",
        "Military-U.S. Army Air Forces","Pan American World Airways","Ameri
can Airlines","Military-Royal Air Force"]
```

In [116]:
```
## Plotting a pie chart:
plt.figure(figsize=(10,7))
plt.pie(dist,autopct="%.2f%%",shadow=True,labels=labels,startangle=20)
plt.title("Percentage Of Crashes Involving The Top 10 Operators Planes")
plt.show()
```



As we can see, Aeroflot has the highest percentage of crashed planes at 24.41%, followed by the U.S. Air Force has the second highest percentage at 24.12%."

Major Reason of Plane Crashes:

In [101]:
```
takeoff=df["Summary"].str.contains("takeoff",case=False,na=False).sum()
```

In [102]:
```
landing=df["Summary"].str.contains("landing",case=False,na=False).sum()
```

In [103]:
```
hijack=df["Summary"].str.contains("hijack",case=False,na=False).sum()
```

In [104]:
```
mistake=df[(df["Summary"].str.contains("error",case=False,na=False))].value
_counts().values.sum()
```

In [105]:
```
weather=df["Summary"].str.contains("weather",case=False,na=False).sum()
```

In [106]:
```
pilot_error=df[(df["Summary"].str.contains("pilot error",case=False,na=Fals
e))].value_counts().values.sum()
```

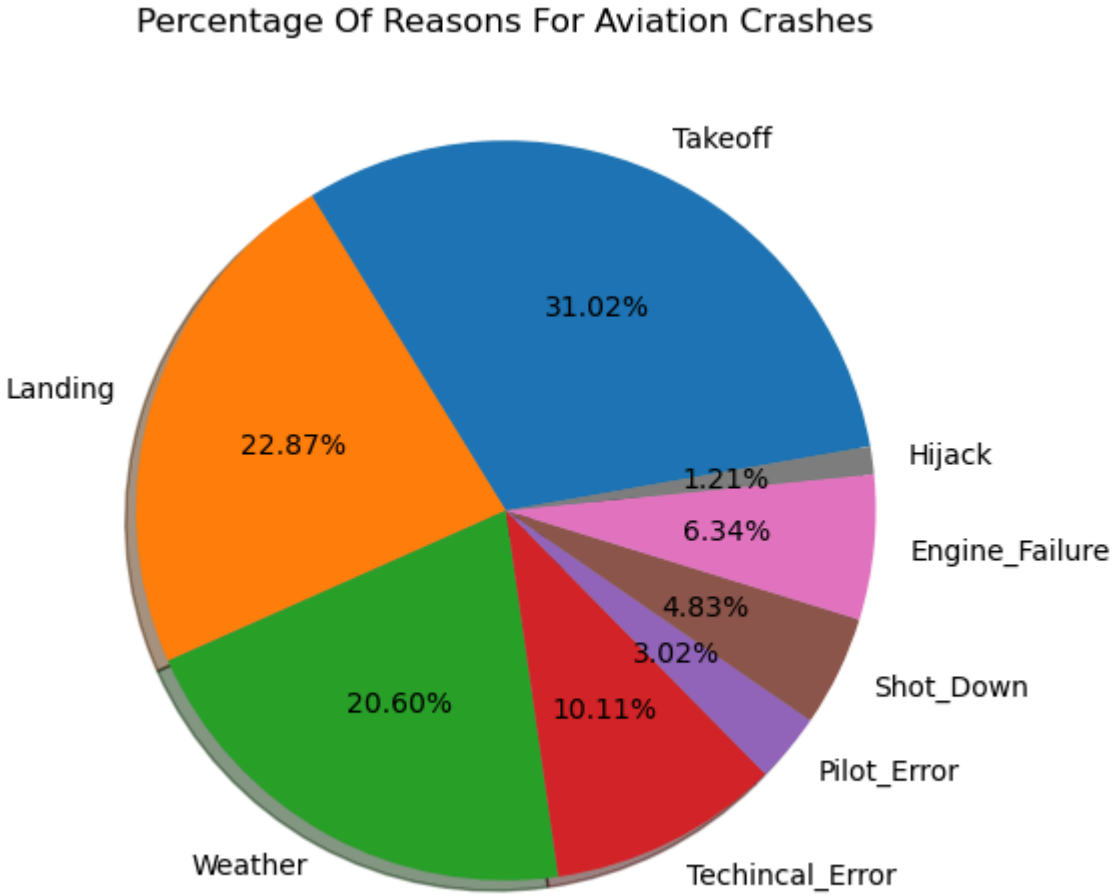
In [107]:
```
shot_down=df["Summary"].str.contains("shot down",case=False,na=False).sum()
```

In [108]:
```
engine_failure=df[(df["Summary"].str.contains("Engine failure",case=False,n
a=False))].value_counts().values.sum()
```

In [109]:
```
chart=[takeoff,landing,weather,mistake,pilot_error,shot_down,engine_failur
e,hijack]
```

In [121]:
```
labels=["Takeoff","Landing","Weather","Techincal_Error","Pilot_Error","Shot
_Down","Engine_Failure","Hijack"]
```

In [122]:
```
plt.figure(figsize=(10,6))
plt.pie(chart,labels=labels,autopct="%.2f%%",shadow=True,startangle=10)
plt.title("Percentage Of Reasons For Aviation Crashes")
plt.show()
```



Conclusions:

The analysis of plane crash data from 1908 to 2008 reveals several critical insights into aviation safety trends and areas needing improvement. The years 1972 and 1968 experienced the highest number of crashes, with 104 and 96 incidents, respectively. Additionally, 1972 recorded the highest number of fatalities, with 2,937 deaths. Fatality rates peaked in 1972 and 1985, with 2,937 and 2,670 fatalities, respectively, reflecting a period of increased risk as passenger numbers grew between 1960 and 2000. Aeroflot and the U.S. Air Force reported the highest crash rates, with 176 and 174 incidents, respectively, and Aeroflot also had the highest fatality count at 7,044, highlighting serious safety concerns. The Douglas DC-3 was notably involved in the most crashes, with 331 incidents and 4,807 fatalities, indicating specific issues with this aircraft model. Training routes were particularly hazardous, accounting for 176 crashes and 940 fatalities, underscoring the need for improved pilot training and safety protocols. Survivors comprised only 21.39% of crash victims, while fatalities accounted for 78.61%, emphasizing the severe nature of these incidents. The year 2001 saw the highest number of hijacking incidents, with 5 occurrences, signaling a critical need for enhanced security measures. Civilian aircraft were involved in 87.09% of crashes, compared to military aircraft at 12.91%, suggesting a continued focus on improving safety in the civilian sector. Overall, while there has been a noticeable decline in crashes since 2000, reflecting advancements in safety technology and practices, the data highlights ongoing challenges and the necessity for continued improvements in pilot training, aircraft technology, and security measures.

In [ ]: